Project

On

“ Publishability and conference Prediction - IIT Kharagpur ”

By

Team – Code Zen

Team Name – Code Zen

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Year | Branch | College |
| Ayush Lokre(Leader) | TY | Artificial Intelligence And Data Science | AISSMS IOIT , Pune |
| Ashutosh Khedkar | TY | Artificial Intelligence And Data Science | AISSMS IOIT , Pune |
| Rohit Kshirsagar | TY | Artificial Intelligence and Data Science | AISSMS IOIT , Pune |
| Vaishnav Kokate | TY | Artificial Intelligence and Data Science | AISSMS IOIT , Pune |

* ABSTRACT

Evaluating research papers for publication and selecting suitable conferences are time-consuming and expert-driven tasks. This project presents an AI-based system to automate these processes by classifying research papers as "Publishable" or "Non-Publishable" and recommending appropriate conferences for publishable papers.

For **Publishability Assessment**, the system uses natural language processing (NLP) techniques to analyze text, extract linguistic features like sentence structure and citations, and generate semantic embeddings with BERT. A Random Forest Classifier, trained on a balanced dataset using SMOTE, ensures accurate classification of papers from various domains.

For **Conference Selection**, publishable papers are matched with leading conferences such as CVPR, EMNLP, and NeurIPS. The system compares the paper’s content with reference papers from these conferences and assigns the best fit with proper justification.

The framework is efficient, scalable, and provides confidence scores with each prediction, helping simplify and speed up the paper evaluation process while maintaining consistency and objectivity. This solution aims to make research workflows more efficient and accessible to the academic community.

* PROBLEM STATEMENT

The manual evaluation of research papers for conference submission is labor intensive, time-consuming, and demands significant expertise. This hackathon challenges participants to develop an AI-driven system using the Pathway Framework to streamline the process of conference selection and research paper evaluation. The system will harness advanced language models, comparative analysis techniques, and streaming data frameworks to automate and optimize these tasks. Participants will have access to a dataset of high-quality, benchmark research papers from conferences or conferences. The objective is to evaluate new submissions, compare them with these benchmark papers, and recommend the most suitable conferences or conference with formal justification.

**Task 1: Research Paper Publishability Assessment**:

* **PDF Feature Extraction**:
  + The extract\_pdf\_features function extracts text and counts the number of pages from research papers.
* **Linguistic Analysis**:
  + compute\_linguistic\_features analyzes sentence structures, named entities, and citation patterns.
* **Transformer-based Embeddings**:
  + encode\_text\_with\_transformer uses **BERT** embeddings to capture semantic features of the text.
* **Classification**:
  + A balanced **RandomForestClassifier** with regularization (e.g., max\_depth, min\_samples\_leaf) predicts the "publishability" of research papers.
* **Imbalance Handling**:
  + **SMOTE** resolves class imbalance during training.
* **Evaluation**:
  + **StratifiedKFold** cross-validation evaluates the classifier's performance on unseen data, ensuring robustness.

**Task 2: Conference Selection**

**Objective:**

Develop a framework to recommend the most suitable conference for a "Publishable" research paper based on:

* Alignment of the paper's content with conference themes and focus areas.
* Benchmark papers provided for reference.
* Use of comparative analysis and content alignment to justify recommendations.

**Requirements:**

1. Benchmark conferences include:
   * CVPR, NeurIPS, DAA, EMNLP, TMLR, KDD
2. Analyze and compare:
   * Subject matter, methodology, and findings.
   * Alignment with conference objectives and standards.
3. Provide a formal justification (up to 100 words) for each recommendation.
4. Use Pathway's streaming capabilities to enhance scalability and real-time performance.

**Implementation in the Code:**

* **Conference Classification**:
  + train\_category\_classifier: Trains a classifier to predict conference categories for publishable papers.
  + predict: Associates publishable papers with conference categories.
* **Justification**:
  + The generate\_rationale method needs to be expanded to include conference-specific explanations.
* PROVIDED SOLUTION

**The solution automates the classification of research papers into "Publishable" and "Non-Publishable" categories, leveraging a combination of Natural Language Processing (NLP), Transformer-based embeddings, and Machine Learning models. For papers classified as publishable, the solution further predicts the most suitable publication category (e.g., CVPR, EMNLP, KDD, NeurIPS, TMLR). By integrating advanced NLP techniques, transformer-based encoding, and robust machine learning methods, the system streamlines the evaluation process and identifies the best-fit conference for publishable papers.**

* **Key Components of the Solution**

**1. Data Processing**

* **Input Data:**

1. **Two datasets are processed:**
   1. **Labeled Papers: Categorized as either publishable or non-publishable.**
   2. **Unlabeled Papers: Papers for which predictions need to be made.**
2. **Input Format:**
   1. **Datasets consist of PDF files.**

* **PDF Feature Extraction:**

1. **Pathway Integration:**
   1. **Pathway's streaming capabilities are used to read and process PDF data from a Google Drive folder.**
   2. **Example: pw.io.gdrive.read() reads the files in a static or streaming mode.**
2. **PyPDF2 is used to extract:**
   1. **Text content from the first three pages of each PDF.**
   2. **Metadata such as the number of pages.**

* **Linguistic Features:**
* **Using spaCy, the following linguistic features are extracted from the text:**
  + **Average Sentence Length: Indicates text complexity.**
  + **Number of Named Entities: Counts identifiable entities (e.g., names, organizations, and locations).**
  + **Number of Citations: Detected using regular expressions matching standard citation formats (e.g., \[\d+\] or (Author et al., Year)).**
  + **Number of Sentences: Measures the overall structure of the paper.**

**2. Text Embedding Using Transformers**

* **Transformer Model:**
  + **A pre-trained BERT (Bidirectional Encoder Representations from Transformers) model is used.**
  + **BERT captures contextual meaning in sentences, generating dense numerical representations (embeddings) for the text.**
  + **Text embeddings are computed using the AutoTokenizer and AutoModel from the Hugging Face Transformers library.**

**3. Feature Combination**

* **Combined Features:**
  + **Numerical features (e.g., number of pages, linguistic features) are concatenated with BERT-based text embeddings using the numpy library.**

**4. Classification Techniques**

* **Publishability Classification:**
* **Model:**
  + **A Random Forest Classifier from scikit-learn is trained to classify papers as publishable (1) or non-publishable (0).**
  + **Regularization parameters like max\_depth and min\_samples\_split are used to prevent overfitting.**
  + **The class\_weight='balanced' parameter ensures the classifier handles imbalanced datasets.**
* **Oversampling with SMOTE:**
* **SMOTE (Synthetic Minority Oversampling Technique):**
  + **Generates synthetic samples for the minority class, balancing the dataset for training.**
* **Category Prediction:**
* **For papers classified as publishable, a secondary Random Forest Classifier predicts the publication category (e.g., CVPR, EMNLP).**

**5. Evaluation**

* **Cross-Validation:**
* **Stratified K-Fold Cross-Validation is used to evaluate classifier performance:**
  + **Splits the dataset into three folds, ensuring class distributions are maintained across splits.**
  + **Computes the F1 score, which balances precision and recall, crucial for imbalanced datasets.**
  + **Reports the mean and standard deviation of F1 scores across folds for robustness.**

**6. Prediction**

* **Unlabeled Dataset:**
* **The trained classifiers:**
  + **Predict whether a paper is publishable or non-publishable.**
  + **For publishable papers, predict the publication category.**
* **Confidence Scores:**
* **Derived from the classifier's prediction probabilities, indicating certainty in predictions.**

**7. Output**

* **Results are saved as a CSV file (results.csv) with the following columns:**
  + **Paper ID: Identifier for each paper.**
  + **Publishable: 1 for publishable, 0 for non-publishable.**
  + **Category: Predicted conference category (e.g., CVPR, EMNLP).**
  + **Rationale: Explanation for the model's decision.**

**▪ Key Python Libraries Used**

1. **Pathway : For streaming and processing data from Google Drive.**
2. **PyPDF2: For extracting text and metadata from PDF files.**
3. **spaCy: For linguistic analysis, including named entity recognition and sentence segmentation.**
4. **Transformers: For generating text embeddings using a pre-trained BERT model.**
5. **numpy: For numerical computations and feature concatenation.**
6. **sklearn: For -** 
   * **Random Forest Classifier.**
   * **Stratified K-Fold Cross-Validation.**
7. **imblearn: For handling class imbalance using SMOTE.**
8. **pandas: For organizing and saving results in a tabular format.**
9. **re: For extracting citations using regular expressions.**

**▪ Model Explanation**

**1. Transformer Model (BERT):**

* **Pre-trained Contextual Understanding:**
  + **Captures contextual meaning in sentences using bidirectional attention.**
  + **Dense embeddings from the last hidden layer represent text features.**

**2. Random Forest Classifier:**

* **Ensemble Learning:**
  + **Combines multiple decision trees for robust predictions.**
  + **Handles imbalanced data with `class\_weight='balanced'.**
  + **Regularized with max\_depth and min\_samples\_split to prevent overfitting.**

**3. SMOTE:**

* **Balances the dataset by synthesizing samples for the minority class (non-publishable papers), ensuring fair training.**

**4. Cross-Validation:**

* **Stratified K-Fold:**
  + **Ensures evaluation across diverse dataset splits while maintaining class distribution.**
  + **F1 score: Measures the balance between precision and recall:**

**▪ Accuracy Calculation**

1. **F1 Score:**
   * **Evaluates performance with imbalanced datasets.**
   * **Reported as the mean across cross-validation folds:**
2. **Confidence Scores:**
   * **Provide probabilities for each prediction, indicating model certainty.**

* **REFERENCES**
* **Pathway Developer Documentation**

**(https://pathway.com/developers/user-guide/introduction/welcome/)**